Exhibit A: Claim Chart

Exhibit A contains Microsoft's Preliminary Claim Construction. The chart presents the construction in the order of the asserted "Mini-Markman" claims. Terms set forth in the claims (column 2) in bold are claim terms that the parties dispute. Phrases set forth in the claims in italics are claim phrases that the parties dispute. Terms set forth in Microsoft's construction (column 3) in bold, with initial capitalizaiton are terms Microsoft has construed.

'193 Asserted Claim 1

	'193 Claim 1	MS Construction
1.	1. A method comprising:	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
2.	*receiving a **digital file** including music,*	receiving a digital file including music: This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed when the **Digital File** is received by Communications Controller 666 and passed through I/O Controller 600 to SPE 503/SPU 500 (specifically incorporates the SPU Encryption/Decryption Engine 522 that is used principally as an aspect of secure communications between **VDE** secure subsystems) and NVRAM 534b (which stores sensitive information such as cryptographic **Key(s)** used for **Authentication.**) Rights Operating System 602 manages the hardware within SPU 500 that performs **Authentication** of the secure container as part of the receiving step. The recited function requires: obtaining a **VDE Secure Container** encapsulating a **Digital File, Authenticating** the intended recipient in accordance with **VDE Controls Associated With** the **Secure Container,** and accepting the **Secure Container.** The qualifier "including music" recites non-functional descriptive material and is not a patentable limitation. digital file: A named, static unit of storage allocated by a "file system" and **Containing** digital information. A **Digital File** enables any application using the "file system" to randomly access its contents and to distinguish it by name from every other such unit. A copy of a **Digital File** is a separate **Digital File.** (A "file system" is the portion of the operating system that translates requests made by application programs for operations on "files" into low-level tasks that can control storage devices such as disk drives.) including: As to data, storing within, as opposed to **Addressing.** As to hardware, physically present within.
3.	storing said **digital file** in a first **secure memory** of a first device;	digital file: see item #2 above secure memory: A processor-addressable **Memory** within a special-purpose **Secure Processing Unit** which is isolated from the rest of the world by (and encapsulated within) a **Tamper Resistant Barrier.** "Processor-addressable" means that a connected processor can use the **Secure Memory**'s physical addresses as the operand in a processor instruction such as LOAD or STORE or equivalent instruction. A "**Memory**" is not a "**Secure Memory**" merely because it stores encrypted, signed, and/or sealed data; is accessible from a **Protected Processing Environment**; or is within an appliance that is located at a trusted facility with non-**VDE** physical **Security** and user-identity **Authentication** procedures.

	'193 Claim 1	MS Construction
		secure: A state in which all users of a system are guaranteed that all information, processes, and devices within the system, shall have their availability, secrecy, integrity, authenticity and nonrepudiation maintained against all of the identified threats thereto. "Availability" means the property that information is accessible and usable upon demand by authorized persons, at least to the extent that no user may delete the information without **Authorization**. "Secrecy," also referred to as confidentiality, means the property that information (including computer processes) is not made available or disclosed to unauthorized persons or processes. "Integrity" means the property that information has not been altered either intentionally or accidentally. "Authenticity" means the property that the characteristics asserted about a person, device, program, information, or process are genuine and timely, particularly as to identity, data integrity, and origin integrity. "Nonrepudiation" means the property that a sender of information cannot deny its origination and that a recipient of information cannot deny its receipt. memory: A medium in which data (including executable instructions) may be stored and from which it may be retrieved.
4.	storing information **associated with said digital file** in a **secure database** stored on said first device,	associated with: A specific, direct, persistent, and binding relationship with one or more discrete items. Code that processes information but is merely a general-purpose component of an installation is not "**Associated With**" that information. In **VDE**, an association between a unit of **Executable** code and particular information, or between particular control information and a **Secure Container**, cannot be broken except as **Allowed** by execution (within a **Secure Processing Environment**) of assigned **VDE Control(s)** and satisfaction of all requirements imposed by such execution. digital file: see item #2 above secure database: A **Secure Database** is a database isolated from all users such that it is **Protected** from external observation; and accidental or intentional alteration or destruction. In **VDE**, a **Secure Database** stores tracking, billing, payment, and auditing data until the data is delivered **Securely** to an authorized **Clearinghouse**. secure: see item #3 above database: a data file that is defined and accessed using the facilities of a database management system (DBMS); this implies in particular (a) that it is defined by means of a schema that is independent of any programs that access the database, and (b) that it uses direct access storage.
5.	said information **including** at least one **budget control** and at least one **copy control**,	including: see item #2 above budget: A unique type of "method" that specifies a decrementable numerical limitation on future **Use** (e.g., copying) of digital information and how such **Use** will be paid for, if at all. (A "method" is a collection of basic instructions, and information related to basic instructions, that provides context, data, requirements, and/or relationships for use in performing, and/or preparing to perform, basic instructions in relation to the operation of one or more electronic appliances.) budget control: A **VDE Control** assembled to apply to a **Budget**, and enforcing that **Budget**. No process, user, or device is able to make the use identified by the **Budget** once the **Budget's** specified limitation on that **Use** has been reached. copy control: A **VDE Control** which **Controls Access** to or some **Use** of a copy.
6.	said at least one **budget control including** ***a budget specifying the number of***	a budget specifying the number of copies which can be made of said digital file: A **Budget** explicitly stating the total number of copies (whether or not decrypted, long-lived, or accessible) that (since creation of the **Budget**) **Can Be** made of the **Digital**

	'193 Claim 1	MS Construction
	copies which can be made of said digital file;	**File** by any and all users, devices, and processes. No process, user, or device is able to make another copy of the **Digital File** once this number of copies has been made. budget, budget control: see item #5 above including: see item #2 above can be: A specified act is able or authorized to be carried out, which otherwise cannot be carried out. digital file: see item #2 above
7.	and said at least one copy control *controlling the copies made of said digital file;*	controlling the copies made of said digital file: **Controlling Uses** of and **Accesses** to all copies of the **Digital File**, by all users, processes, and devices, by executing each of the recited "at least one" **Copy Control(s)** within **VDE Secure Processing Environment(s)**. Each **Control Governs (Controls)** only one action, which action may or may not differ among the different "at least one" **Controls**. All **Uses** and **Accesses** are prohibited and incapable of occurring except to the extent **Allowed** by the "at least one" **Copy Control(s)**. copy control: see item #5 above controlling: Reliably defining and enforcing the conditions and requirements under which an action that otherwise cannot be taken, will be **Allowed**, and the manner in which it may occur. Absent verified satisfaction of those conditions and requirements, the action cannot be taken by any user, process or device. In **VDE**, an action is **Controlled** through execution of the applicable **VDE Control(s)** within a **VDE Secure Processing Environment**. More specifically, in **VDE**, **Controlling** is effected by use of **VDE Controls**, **VDE Secure Containers**, and **VDE** foundation (including **VDE Secure Processing Environment**, "object registration," and other mechanisms for allegedly individually ensuring that specific **Controls** are enforced vis-à-vis specific objects (and their content at an arbitrary granular level) and specific "users.") digital file: see item #2 above
8.	*determining whether said digital file may be copied and stored on a second device based on at least said copy control;*	determining whether said digital file may be copied and stored on a second device based on at least said copy control: Determining whether this particular first device is **Allowed** to perform both of the following actions on this particular **Digital File**: (1) **Copy** it and (2) store it (as opposed to a copy of it) on a second device, by executing one or more **VDE Control(s)** (including "said" **Copy Control Associated With** this **Digital File**) within **VDE Secure Processing Environment**(s). To the extent that either of these two actions is not determined by this step to be permissible, that action is prohibited and incapable of occurring, and no user, process or device can perform it on this **Digital File**. This claim limitation's recitation of "said copy control" is inconsistent with the claim limitation "at least one copy control." digital file: see item #2 above copy, copied, copying: To reproduce all of a **Digital File** or other complete physical block of data from one location on a storage medium to another location on the same or different storage medium, leaving the original block of data unchanged, such that two distinct and independent objects exist. Although the layout of the data values in physical storage may differ from the original, the resulting "copy" is logically indistinguishable from the original. The resulting "copy" may or may not be encrypted, ephemeral, usable, or accessible.

	'193 Claim 1	MS Construction
		copy control: see item #5 above
9.	*if said copy control allows at least a portion of said digital file to be copied and stored on a second device,*	if said copy control allows at least a portion of said digital file to be copied and stored on a second device: This "if" condition creates two branches for the recited process, each of which must be performed. Each time the "if" condition is met, all four of the later-recited actions (**Copying**, transferring, storing, playing) must occur. Each time it is not met, each of these four actions must be prohibited and incapable of occurring. This "if" condition is met if and only if "said" **Copy Control Allows** any **Portion** (i.e., a part less than the whole) of the **Digital File** to be **Copied** and also **Allows** that same **Portion** of the **Digital File** (as opposed to the copy) to be stored on any second device. This "if" condition is based entirely on "said copy control" and thus is met, as above, even if other **VDE Control(s)** prohibit those actions. This claim limitation's recitation of "copy control allows at least a portion" is inconsistent with the claim limitation "whether said digital file may be copied ... based on at least said copy control." This claim limitation's recitation of "if said copy control allows at least a portion ... copying" is inconsistent with "said at least one budget control including a budget specifying the number of copies which can be made of said digital file" on whether said "copy control" or said "budget control" determines whether **Copying** is **Allowed**. copy control: see item #5 above allow (allows): Actively permitting an action that otherwise cannot be taken (i.e., is prohibited) by any user, process, or device. In **VDE**, an action is **Allowed** only through execution (within a **Secure Processing Environment**) of the **VDE Control(s)** assigned to the particular action request, and satisfaction of all requirements imposed by such execution. portion: A part of a whole, which is less than the whole digital file: see item #2 above
10.	*copying at least a portion of said digital file;*	copying at least a portion of said digital file: **Copying** at least some **Portion** of the **Digital File** (as opposed to a copy thereof), by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. This **Copied "Portion"** may or may not be (or even include) the **Portion** referred to in the claim limitation "if said copy control allows at least a portion." copying: see item #8 above portion: see item #9 above digital file: see item #2 above
11.	*transferring at least a portion of said digital file to a second device* including a **memory** and an audio and/or video output;	transferring at least a portion of said digital file to a second device: Transferring to some second device (which may or may not be the "second device" referred to in the claim limitation "if said copy control allows at least a portion of said digital file to be copied and stored on a second device") at least some **Portion** of the **Digital File** (as opposed to a copy thereof), by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. This transferred **Portion** may or may not be (or even include) the **Portion** referred to in the claim limitation "if said copy control allows at least a portion," or the **Portion** referred to in the claim limitation "copying at least a portion."

	'193 Claim 1	MS Construction
		portion: see item #9 above digital file: see item #2 above memory: see item #3 above
12.	*storing said digital file* in said **memory** of said second device; and	storing said digital file: Storing the entire **Digital File** received in the "receiving" step (as opposed to a copy of the **Digital File** or a **Portion** of the **Digital File**). This claim limitation's recitation of "storing said digital file" is inconsistent with the claim limitation "transferring at least a portion of said digital file." digital file: see item #2 above memory: see item #3 above
13.	including playing said music through said audio output.	This claim limitation's recitation of "playing ... through said audio output" is inconsistent with the claim limitation "an audio and/or video output."

	'193 - Claim 11	MS Construction
14.	11. A method comprising:	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
15.	*receiving a **digital file***	receiving a digital file: see item #2 above digital file: see item #2 above
16.	storing information **associated with** said **digital file** in a **secure database** stored on said first device,	associated with: see item #4 above digital file: see item #2 above secure database: see item #4 above
17.	said information **including** a first **control**;	including: see item #2 above control: Independent, special-purpose, **Executable**, which can execute only within a **Secure Processing Environment**. Each **VDE Control** is a **Component Assembly** dedicated to a particular activity (e.g., editing, modifying another **Control**, a user-defined action, etc.), particular user(s), and particular **Protected** information, and whose satisfactory execution is necessary to **Allowing** that activity. Each separate information **Access** or **Use** is independently **Controlled** by independent **VDE Control(s)**. Each **VDE Control** is assembled within a **Secure Processing Environment** from independently deliverable modular components (e.g., **Load Modules** or other **Controls**), dynamically in response to an information **Access** or **Use** Request. The dynamic assembly of a **Control** is directed by a "blueprint" **Record** (put in place by one or more **VDE** users) **Containing** control information identifying the exact modular code components to be assembled and executed to **Govern** this particular activity on this particular information by this particular user(s). Each **Control** is independently assembled, loaded and delivered vis-à-vis other **Controls**. Control information and **Controls** are extensible and can be configured and modified by all users, and combined by all users with any other **VDE Control** information or **Controls** (including that provided by other users), subject only to "senior" user **Controls**. Users can assign control information (including alternative control information) and controls to an arbitrarily fine, user-defined **Portion** of the **Protected** information, such as a single paragraph of a document, as opposed to being limited to file-based **Controls**. **VDE Controls** reliably limit **Use** of the **Protected** information to **Authorized** activities and amounts.
18.	*determining whether said **digital file** may be **copied** and stored on a second device based on said first **control**,*	determining whether said digital file may be copied and stored on a second device based on said first control: Determining whether said first **Control**, by itself, **Allows** this particular first device to perform both of the following actions on this particular **Digital File**: (1) **Copy** it and (2) store it (as opposed to a copy of it) on a second device, by executing the first **VDE Control** within **VDE Secure Processing Environment(s)**. To the extent that either the **Copy** or store action is not determined by this step to be permissible, that action is prohibited and incapable of occurring, and no user, process or device can perform it on this **Digital File**. digital file: see item #2 above copied: see item #10 above control: see item #17 above
19.	said determining step including *identifying said second device* and	identifying said second device: **Identifying** a second device sufficiently to distinguish it from all other devices, by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**.

	'193 - Claim 11	MS Construction
	determining *whether said first control allows transfer of said copied file to said second device,*	whether said first control allows transfer of said copied file to said second device Whether the first **Control**, by itself, **Allows** the entire **Digital File** (which has been **Copied** at least once) (as opposed to the copy) to be moved to the identified second device. If not, that transfer is prohibited and incapable of occurring and no user, process or device can perform that action on this **Digital File**. Identifying/identify: To establish as being a particular instance of a person or thing control: see item #17 above allow: see item #9 above copied file: A **Digital File** that has been **Copied**. The "copied file" is not the copy itself. A "copy" is what is formed by a **Copying** operation, and it may or may not be encrypted, ephemeral, usable, or accessible.
20.	*said determination based at least in part on the features present at the device* to which said **copied file** is to be transferred;	said determination based at least in part on the features present at the device: Basing the determination at least in part upon all actual, current features of the device (as opposed to previously determined, reported, or measured features) which might affect the device's ability to prevent **Unauthorized Access** to or **Use** of (or both) the **Digital File**. This determination is done without trusting either the device or any user of the device. A device **Identifier** such as a serial number is not a "feature present at the device." copied file: see item #19 above
21.	*if said first control allows at least a portion of said digital file to be copied and stored on a second device,*	if said first control allows at least a portion of said digital file to be copied and stored on a second device: This "if" condition creates two branches for the recited process, each of which must be performed. Each time the "if" condition is met, all four of the later-recited actions (**Copying**, transferring, storing, **Rendering**) must occur. Each time it is not met, each of these four actions must be disabled and prohibited and incapable of occurring. This "if" condition is met if and only if the first **Control** allows any **Portion** of the **Digital File** to be **Copied** and also allows that same **Portion** of the **Digital File** (as opposed to the copy) to be on any second device. This "if" condition is based entirely on the first **Control** and thus is met, as above, even if other **VDE Controls** prohibit those actions. This claim limitation's recitation of "said first control allows at least a portion" is inconsistent with the claim limitation "whether said digital file may be copied ... based on said first control." control: see item #17 above allow: see item #9 above portion: see item #9 above digital file: see item #2 above
22.	*copying at least a portion of said digital file;*	copying at least a portion of said digital file: see item #10 above copying: see item #8 above portion: see item #9 above

	'193 - Claim 11	MS Construction
		digital file: see item #2 above
23.	*transferring at least a portion of said digital file to a second device* including a **memory** and an audio and/or video output;	transferring at least a portion of said digital file to a second device: see item #11 above portion: see item #9 above digital file: see item #2 above memory: see item #3 above
24.	*storing said digital file* in said memory of said second device; and	storing said digital file: see item #12 above digital file: see item #2 above
25.	**rendering** said digital file through said output.	rendering: Playing content through an audio output (e.g., speakers) or displaying content on a video output (e.g., a screen). digital file: see item #2 above This claim limitation's recitation of "said output" is inconsistent with the claim limitation "an audio and/or video output."

	'193 Claim 15	MS Construction
26.	15. A method comprising:	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
27.	*receiving a digital file;*	receiving a digital file: see item #2 above This step must proceed in both "**Authentication** branches" of the process (i.e., regardless of the outcome of the "**Authentication**" step). digital file: see item #2 above
28.	*an authentication step comprising:*	an authentication step comprising: **Authenticating** the first device and/or user of the first device without relying on trusting either, by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. authentication: To establish that the following asserted characteristics of something (e.g., a person, device, organization, document, file, etc.) are genuine: its **Identity**, its data integrity, (i.e., it has not been altered) and its origin integrity (i.e., its source and time of origination).
29.	*accessing at least one identifier associated with a first device or with a user of said first device;* and	accessing at least one identifier associated with a first device or with a user of said first device: **Securely Accessing** at least one **Identifier Associated With** a single ("first") device or (as opposed to "and") with a single, current user of that device, by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. One of the "at least one identifier" may be **Associated With** a first device while another of the "at least one identifier" may be **Associated With** a user of said first device. Access (accessing): To satisfactorily perform the steps necessary to obtain something so that it can be **Used** in some manner (e.g., for information: copied, printed, decrypted, encrypted, saved, modified, observed, or moved, etc.). In **VDE**, access to protected information is achieved only through execution (within a **Secure Processing Environment**) of the **VDE Control(s)** assigned to the particular "access" request, satisfaction of all requirements imposed by such execution, and the **Controlled Opening** of the **Secure Container Containing** the information. identifier: Any text string used as a label naming an individual instance of what it **Identifies**. associated with: see item #4 above
30.	*determining whether said identifier is associated with a device and/or user authorized to store said digital file;*	determining whether said identifier is associated with a device and/or user authorized to store said digital file: For each accessed "at least one identifier," determining whether the device with which it is **Associated** is one on which the **Digital File** may be stored (by any user) and/or whether the user with which it is **Associated** is one who may store the **Digital File** (on any device), by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. Each **Identifier** may be **Associated With** a device "and" a user, or with a device only, or with a user only. This claim limitation's recitation of "said identifier" is inconsistent with the claim limitation "at least one identifier." identifier: see item #29 above associated with: see item #4 above authorized: An action is permitted that otherwise cannot be taken by any user, process, or device. In **VDE**, an action is authorized only through execution of the applicable **VDE Control(s)** within a **VDE Secure Processing Environment** and

	'193 Claim 15	MS Construction
		satisfaction of all requirements imposed by such execution. "not authorized": The action is prohibited and cannot be taken by any user, process, or device. digital file: see item #2 above
31.	*storing said **digital file** in a first **secure memory** of said first device, but only if said device and/or user is so **authorized**, but not proceeding with said storing if said device and/or user is not **authorized**;*	storing said digital file in a first secure memory of said first device, but only if said device and/or user is so authorized, but not proceeding with said storing if said device and/or user is not authorized: This conditional step creates at least two "**Authentication**" branches for the recited process, each of which must be performed. Each time the condition is met, the recited "storing" must occur. Each time it is not met, the recited "storing" must not occur. If "storing" proceeds, then: storing in a **Secure Memory** of the first device, the entire **Digital File** received in the "receiving" step, as opposed to a copy of the **File** or a **Portion** of the **Digital File**, by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. If "storing" does not proceed: then the **Digital File** is not stored in the **Secure Memory** of the first device, and is prevented from being stored anywhere on the first device. This limitation is internally inconsistent on the circumstances under which the storing proceeds or does not proceed. For example, the first ("only if") phrase requires that the storing step proceeds if the device is **Authorized** (and the user is not) while the second ("but not") phrase requires that the storing step not proceed if the device is **Authorized** (and the user is not). authorized: see item #30 above digital file: see item #2 above secure memory: see item #3 above
32.	*storing information **associated with said digital file** in a **secure database** stored on said first device, said information including at least one **control**;*	storing information associated with said digital file in a secure database stored on said first device, said information including at least one control: Storing information in a **Secure Database**, the entirety of information (including the "at least one Control") being **Associated With** the **Digital File** (as opposed to the file's contents independent of the file), by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. This step must proceed in both "**Authentication** branches" of the process (i.e., regardless of the outcome of the "**Authentication**" step). associated with: see item #4 above digital file: see item #2 above secure database: see item #4 above control: see item #17 above
33.	*determining whether said **digital file** may be **copied** and stored on a second device based on said at least one **control**;*	determining whether said digital file may be copied and stored on a second device based on said at least one control: see item #8 above This step must proceed in both "**Authentication** branches" of the process (i.e., regardless of the outcome of the "**Authentication**" step). digital file: see item #2 above

	'193 Claim 15	MS Construction
		copied: see item #10 above control: see item #17 above
34.	*if said at least one control allows at least a portion of said digital file to be copied and stored on a second device,*	if said at least one control allows at least a portion of said digital file to be copied and stored on a second device: see item #9 above control: see item #17 above allow: see item #9 above portion: see item #9 above digital file: see item #2 above copied: see item #10 above
35.	*copying at least a portion of said digital file;*	copying at least a portion of said digital file: see item #10 above copying: see item #8 above portion: see item #9 above digital file: see item #2 above
36.	*transferring at least a portion of said digital file to a second device* including a **memory** and an audio and/or video output;	transferring at least a portion of said digital file to a second device: see item #11 above This step must proceed in both "**Authentication** branches" of the process (i.e., regardless of the outcome of the "**Authentication**" step). portion: see item #9 above digital file: see item #2 above memory: see item #3 above
37.	*storing said digital file* in said **memory** of said second device; and	storing said digital file: see item #12 above This step must proceed in both "**Authentication** branches" of the process (i.e., regardless of the outcome of the "**Authentication**" step). This claim limitation's recitation of "storing said digital file" is inconsistent with the claim limitation "transferring at least a portion of said digital file." digital file: see item #2 above memory: see item #3 above
38.	rendering said digital file through said output.	rendering: see item #25 above digital file: see item #2 above This claim limitation's recitation of "said output" is inconsistent with the claim limitation "an audio and/or video output."

	'193 Claim 19	MS Construction
39.	19. A method comprising:	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
40.	*receiving a **digital file** at a first device;*	receiving a digital file at a first device: see item #2 above digital file: see item #2 above
41.	*establishing communication between said first device and a **clearinghouse** located at a location remote from said first device;*	establishing communication between said first device and a clearinghouse located at a location remote from said first device: This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("establishing communication") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step Part of the recited function is performed by the Remote Procedure Call Manager 732 software of Rights Operating System 602 that controls I/O controller 660 and Communications Controller 666. Remote Procedure Call Manager handles all communication between **VDE** processes. The recited function is: creating and using a previously non-existent communications channel which is necessary and sufficient for exchanging information between the first device and a **Clearinghouse**. clearinghouse: A computer system that provides intermediate storing and forwarding services for both content and audit information, and which two or more parties trust to provide its services independently because it is operated under constraint of **VDE Security**. "Audit information" means all information created, stored, or reported in connection with an "auditing" process. "Auditing" means tracking, metering and reporting the usage of particular information or a particular appliance.
42.	said first device obtaining **authorization information** including a **key** from said clearinghouse;	authorization information: "Control information" identifying the exact modular code components to be assembled into a **VDE Control** and executed within a **Secure Processing Environment** to permit a particular activity that otherwise cannot be taken (i.e., is prohibited). ("Control information" is information which **Identifies** the exact modular code components and data which must be assembled and executed to **Control** a particular activity on particular information, of arbitrary, user-defined granularity, by particular user(s)). key: A bit sequence used and needed by a cryptographic algorithm to encrypt a block of plain text or to decrypt a block of cipher text. A **Key** is different from a key seed or other information from which the actual encryption and/or decryption **Key** is constructed, derived, or otherwise identified. In symmetric key cryptography, the same key is used for both encryption and decryption. In asymmetric or "public key" cryptography, two related keys are used; a block of text encrypted by one of the two keys (e.g., the "public key") can be decrypted only by the corresponding key (e.g., the "private key"). clearinghouse: see item #41 above
43.	said first device *using said **authorization information** to gain access to or make at least one use of said first **digital file**,*	using said authorization information to gain access to or make at least one use of said first digital file: A user, process or device uses all of said **Authorization Information** in connection with executing **VDE Control(s)** within **VDE Secure Processing Environment(s)** to gain **Access** to or (as opposed to "and") make at least one **Use** of the **Digital File** received in the "receiving" step. Without using such **Authorization Information**, no **Access** to or **Use** of the file is **Allowed**. authorization information: see item #42 above access: see item #29 above

	'193 Claim 19	MS Construction
		use: To use information is to perform some action on it or with it (e.g., copying, printing, decrypting, encrypting, saving, modifying, observing, or moving, etc.). In **VDE**, information **Use** is **Allowed** only through execution of the applicable **VDE Control(s)** and satisfaction of all requirements imposed by such execution. digital file: see item #2 above
44.	*including using said key to decrypt at least a portion of said first digital file*; and	including using said key to decrypt at least a portion of said first digital file: The "at least one use of said digital file" must encompass decrypting at least a **Portion** of the **Digital File** using the **Key**. portion: see item #9 above digital file: see item #2 above
45.	*receiving a first control from said clearinghouse at said first device;*	receiving a first control from said clearinghouse at said first device: This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b). The recited function requires: obtaining a **VDE Secure Container** encapsulating a first **Control**, authenticating the first device in accordance with **VDE Controls Associated With the Secure Container**, and accepting the **Secure Container**. control: see item #17 above clearinghouse: see item #41 above
46.	*storing said first digital file in a memory of said first device;*	storing said first digital file in a memory of said first device: Storing in a **Memory** of the first device, the entire **Digital File** (as opposed to a **Portion** thereof) received in the "receiving" step, by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. digital file: see item #2 above memory: see item #3 above
47.	*using said first control to determine whether said first digital file may be copied and stored on a second device;*	using said first control to determine whether said first digital file may be copied and stored on a second device: Determining whether the first **Control**, by itself, allows this particular first device to perform both of the following actions on this particular **Digital File**: (1) **Copy** it and (2) store it (as opposed to a copy of it) on a second device, by executing the first **VDE** Control within **VDE Secure Processing Environment(s)**. To the extent that either the **Copy** or store action is not determined by this step to be permissible, that action is prohibited and incapable of occurring, and no user, process or device can perform it on this **Digital File**. control: see item #17 above digital file: see item #2 above copied: see item #10 above
48.	*if said first control allows at least a portion of said*	if said first control allows at least a portion of said first digital file to be copied and stored on a second device: see item #9 above

	'193 Claim 19	MS Construction
	first digital file to be copied and stored on a second device,	This claim limitation's recitation of "first control allows at least a portion of said first digital file" is inconsistent with the claim limitation "whether said first digital file may be copied ... on a second device." control: see item #17 above allow: see item #9 above portion: see item #9 above digital file: see item #2 above copied : see item #10 above
49.	*copying at least a portion of said first digital file;*	copying at least a portion of said first digital file: see item #10 above copying: see item #8 above portion: see item #9 above digital file: see item #2 above
50.	*transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output;*	transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output: see item #11 above portion: see item #9 above digital file: see item #2 above memory: see item #3 above
51.	*storing said first digital file portion* in said memory of said second device; and	storing said first digital file portion: Storing the "at least a portion" which was transferred to the second device, of the **Digital File** received in the "receiving" step (as opposed to a copy of the **Digital File**). digital file: see item #2 above portion: see item #9 above memory: see item #3 above
52.	rendering said first digital file portion through said output.	rendering: see item #25 above portion: see item #9 above digital file: see item #2 above This claim limitation's recitation of "said output" is inconsistent with the claim limitation "an audio and/or video output."

	'683 Claim 2	MS Construction
53.	2. A system including:	Claim as a Whole: The "system" is a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
54.	a first apparatus including,	
55.	**user controls,**	user controls: **Controls** created, modified, or selected by a user to **Control** a particular **Use** or **Access** by the user to particular **Protected** information. control: see item #17 above
56.	a communications port,	
57.	a processor,	
58.	a **memory** storing:	memory: see item #3 above
59.	a first **secure container**	secure container: A **VDE Secure Container** is a self-contained, self-protecting data structure which (a) encapsulates information of arbitrary size, type, format, and organization, including other, nested, containers, (b) cryptographically protects that information from all unauthorized **Access** and **Use**, (c) provides encrypted storage management functions for that information, such as hiding the physical storage location(s) of its protected contents, (d) permits the **Association** of itself or its contents with **Controls** and **Control** information **Governing Access** to and **Use** thereof, and (e) prevents such **Use** or **Access** (as opposed to merely preventing decryption) until it is "opened." A **Secure Container** can be opened only as expressly **Allowed** by the associated **VDE Control(s)**, only within a **Secure Processing Environment**, and only through decryption of its encrypted header. A **Secure Container** is not directly accessible to any non-**VDE** or user calling process. All such calls are intercepted by **VDE**. The creator of a **Secure Container** can assign (or allow others to assign) control information to any arbitrary **Portion** of a **Secure Container's** contents, or to an empty **Secure Container** (to **Govern** the later addition of contents to the container, and **Access** to or **Use** of those contents). A container is not a **Secure Container** merely because its contents are encrypted and signed. A **Secure Container** is itself **Secure**. All **VDE-Protected** information (including protected content, information about content usage, content-control information, **Controls**, and **Load Modules**) is encapsulated within a **Secure Container** whenever stored outside a **Secure Processing Environment** or **Secure Database**.
60.	**containing a governed item,**	containing: Physically (directly) storing within, as opposed to Addressing. governed item: Information, of arbitrarily fine granularity, whose **Access** and **Use** by any user, process, or device is **Controlled**.
61.	the first **secure container governed item** being at least in part encrypted;	secure container: see item #59 above governed item: see item #60 above
62.	*the first secure container having been received from a second apparatus;*	the first secure container having been received from a second apparatus: The "first secure container" must **Identify** the single apparatus from which it was received, and that apparatus must be different from the first apparatus. Alternatively, if the Court does not construe this claim language as requiring the "first secure container" to identify the single apparatus from which it was received: This claim language has no patentable weight. It recites a step taken in the creation of the recited system, not a structural or functional characteristic of the system. One studying a particular system (as opposed to the process by which it was created) to compare it to the claimed system, could not distinguish a **Secure Container** received from another apparatus from, e.g., a **Secure Container** created on the first apparatus, and thus could not determine whether this step was satisfied.

	'683 Claim 2	MS Construction
		Receiving the **Secure Container** includes **Authenticating** the intended recipient in accordance with **VDE Controls Associated With** the **Secure Container**. The first **Secure Container** may be received as bar codes in a fax transmission, or filled ovals on a form delivered through physical mail. secure container: see item #59 above
63.	a first **secure container rule**	secure container rule: A Rule that **Governs** a **Secure Container Governed Item**. rule: A lexical statement that states a condition under which **Access** to or **Use** of **VDE-Protected** data will be **Allowed** by a **VDE Control**. A rule may specify how, when, where, and by whom a particular activity on particular information is to be **Allowed**.
64.	at least in part **governing** *an aspect of access to or use of* said first secure container governed item,	an aspect of access to or use of: Any one (as opposed to more than one) aspect of any **Access** to or (as opposed to "and") **Use** by any and all processes, users, and devices. governing: see Control (v.) item #7 above aspect: An aspect of an environment is a persistent element or property of that environment that can be used to distinguish it from other environments. access: see item #29 above use: To use information is to perform some action on it or with it (e.g., copying, printing, decrypting, encrypting, saving, modifying, observing, or moving, etc.). In **VDE**, information **Use** is **Allowed** only through execution of the applicable **VDE Control(s)** and satisfaction of all requirements imposed by such execution.
65.	the first **secure container rule**, *the first secure container rule having been received from a third apparatus different from said second apparatus;* and	the first secure container rule having been received from a third apparatus different from said second apparatus: The "first secure container rule" must have been received encapsulated within a **VDE Secure Container**, and the intended recipient must have been **Authenticated** in accordance with **VDE Controls Associated With** the **Secure Container**, and the "first secure container rule" must have been accepted by the first apparatus. The "first secure container rule" must identify the single apparatus from which it was received, and that apparatus must be different from the first apparatus. Alternatively, if the Court does not construe this claim language as requiring the "first secure container" to identify the single apparatus from which it was received: This claim language has no patentable weight. It recites a step taken in the creation of the recited system, not a structural or functional characteristic of the system. One studying a particular system (as opposed to the process by which it was created) to compare it to the claimed system, could not distinguish a **Secure Container Rule** received from another apparatus from, e.g., a **Secure Container Rule** created on the first apparatus, and thus could not determine whether this step was satisfied. secure container rule: see item #63 above
66.	*hardware or software used for receiving and opening secure containers,*	hardware or software used for receiving and opening secure containers, receiving: This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites an undefined mechanism ("hardware or software") for performing a function (e.g., "Opening") without reciting particular structure that performs that function. The specification does not clearly link any particular structure to this recited function. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b). The recited function requires: the same single logical piece of either hardware or software (as opposed to both) must be capable of both receiving and **Opening Secure**

	'683 Claim 2	MS Construction
		Containers, this "receiving" including authenticating the intended recipient in accordance with **VDE Controls Associated With the Secure Container**, and this "**Opening**" performed by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. opening secure containers: Establishing the requisites needed to attempt to access the contents of a **Secure Container**. Opening is a necessary but insufficient step before the contents of a **Secure Container** may be copied, decrypted, read, manipulated, or otherwise **Used**, or **Accessed**. No process, user, or device may **Access** or **Use** the contents of a **Secure Container** without first opening that **Secure Container**. A **Secure Container** may be opened only through execution of the assigned **VDE Control(s)** within a **VDE Secure Processing Environment** and satisfaction of all requirements imposed by such execution.
67.	*said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;*	said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers: Each **Secure Container** referred to in the phrase "hardware or software used for receiving and opening secure containers" must have the capacity to Contain a **Governed Item**, and must have **Associated With** it a **Secure Container Rule**. By "each secure container referred to in the phase ...," is meant each **Secure Container** which the "hardware or software used for receiving and opening secure containers" is capable of receiving and **Opening**. The **Secure Container Rule** is **Associated With** the **Secure Container** itself, as opposed to a **Governed Item**. secure container: see #59 above capacity: Available storage space that is still capable of allocation. For example, a 650 MB blank CD, after sealing, has zero capacity because no new material may be stored within it. contain: see item #60 above governed item: see item #60 above secure container rule: see item #63 above associated with: see item #4 above
68.	a *protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus,*	protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus: A single **VDE Secure Processing Environment**, in addition to and not within the first apparatus, actively **Preventing** (not merely being capable of **Preventing**, and not merely resisting) any "user" of the first apparatus from **Tampering** with any and all information encapsulated by the **Secure Processing Environment** (as opposed to **Tampering** with the **Secure Processing Environment** itself). Other components may or may not provide part of this **Protecting** function. The **Protecting** function is provided by use of the disclosed "Component Assembly" (**VDE Controls**), "Secure Container," "Protected Processing Environment," "object registration," and other mechanisms of the purported "VDE" "invention" for allegedly individually ensuring the "Access Control" "handcuffs" between specific "Controls," specific "objects" (and their content at an arbitrary granular level), and specific "users." protected processing environment: A uniquely identifiable, self-contained computing base trusted by all **VDE** nodes to protect the availability, secrecy, integrity and authenticity of all information identified in the February, 1995, patent application as being protected, and to guarantee that such information will be accessed and used only as expressly authorized by **VDE Controls**. At most **VDE** nodes, the **Protected**

	'683 Claim 2	MS Construction
		Processing Environment is a **Secure Processing Environment** which is formed by, and requires, a hardware **Tamper Resistant Barrier** encapsulating a special-purpose Secure Processing Unit having a processor and internal secure **Memory**. ("Encapsulated" means hidden within an object so that it is not directly accessible but rather is accessible only through the object's restrictive interface.) The barrier prevents all unauthorized (intentional or accidental) interference, removal, observation, and Use of the information and processes within it, by all parties (including all users of the device in which the **Protected Processing Environment** resides), except as expressly authorized by **VDE Controls**. A **Protected Processing Environment** is under **Control** of **Controls** and control information provided by one or more parties, rather than being under **Control** of the appliance's users or programs. Where a **VDE** node is an established financial **Clearinghouse**, or other such facility employing physical facility and user-identity **Authentication Security** procedures trusted by all **VDE** nodes, and the **VDE** node does not **Access** or use **VDE-protected** information, or assign **VDE** control information, then the **Protected Processing Environment** at that **VDE** node may instead be formed by a general-purpose CPU that executes all **VDE** "security" processes in **Protected** (privileged) mode. A **Protected Processing Environment** requires more than just verifying the integrity of **Digitally Signed Executable** programming prior to execution of the programming; or concealment of the program, associated data, and execution of the program code; or use of a password as its protection mechanism. protecting: Maintaining the **Security** of. contain (contained): see item #60 above
69.	said protected processing environment **including** *hardware or software used for applying said first **secure container rule** and a second secure container rule **in combination** to at least in part govern at least one **aspect** of **access** to or use of a **governed item** contained in a **secure container***; and	hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container: This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites an undefined mechanism ("hardware or software") for performing a function ("applying ... in combination") without reciting particular structure that performs that function. The specification does not clearly link any particular structure to this recited function. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b). The recited function requires: a single logical piece of either hardware or software (as opposed to both) to apply the two separate **Rules** in combination by assembling and executing a single **Control**, and to **Govern** any one or more aspects of any **Access** or **Use** by any process or user or device, of a **Governed Item Contained** in a **Secure Container** (which may or may not be any "Secure Container" recited earlier). Other components may or may not provide part of the **Governing** function. This "hardware or software" performs its functions by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. including: see item #2 above aspect: see item #64 above access: see item #29 above contain (contained): see item #60 above secure container rule: see item #63 above secure container: see #59 above

	'683 Claim 2	MS Construction
		governed item: see item #60 above
70.	*hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.*	hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses: This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites an undefined mechanism ("hardware or software") for performing a function (e.g., "transmission") without reciting particular structure that performs that function. The specification does not clearly link any particular structure to this recited function. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b). The recited function requires: a single logical piece of either hardware or software (as opposed to both) is capable of both transmission and receipt of **Secure Containers**, this receipt including **Authenticating** the intended recipient in accordance with **VDE Controls Associated With the Secure Container**. This "hardware or software" is separate from and in addition to the first apparatus, the recited "protected processing environment," and the recited "hardware or software used for receiving and opening secure containers." The transmission and receipt of the **Secure Containers** may be via bar codes in a fax transmission, or filled ovals on a form delivered through physical mail. This "hardware or software" performs its functions by executing **VDE Control(s)** within **VDE Secure Processing Environment(s)**. secure container: see #59 above

	'721 Claim 1	MS Construction
71.	1. A security method comprising:	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
72.	***digitally signing a first load module with a first digital signature designating the first load module for use by a first device class;***	digitally signing a first load module with a first digital signature designating the first load module for use by a first device class: **Digitally Signing** a particular ("first") **Load Module** by using a first **Digital Signature** as the signature **Key**, which signing indicates to any and all devices in the first device class that the signor authorized and restricted this **Load Module** for **Use** by that device. No **VDE** device can perform any execution of any **Load Module** without such authorization. The method ensures that the **Load Module** cannot execute in a particular device class and ensures that no device in that device class has the **Key(s)** necessary to verify the **Digital Signature**. digital signature: digital signature: A computationally unforgeable string of characters (e.g., bits) generated by a cryptographic operation on a block of data using some secret. The string can be generated only by an Entity that knows the secret, and hence provides evidence that the Entity must have generated it. digitally signing: Creating a **Digital Signature** using a secret **Key**. (In symmetric key cryptography, a "secret key" is a **Key** that is known only to the sender and recipient. In asymmetric key cryptography, a "secret key" is the private **Key** of a public/private key pair, in which the two keys are related uniquely by a predetermined mathematical relationship such that it is computationally infeasible to determine one from the other.) load module: An **Executable**, modular unit of machine code (which may include data) suitable for loading into **Memory** for execution by a processor. A **Load Module** is encrypted (when not within a secure processing unit) and has an **Identifier** that a calling process must provide to be able to use the **Load Module**. A **Load Module** is combinable with other **Load Modules**, and associated data, to form **Executable Component Assemblies**. A **Load Module** can execute only in a **VDE Protected Processing Environment**. Library routines are not **Load Modules** and dynamic link libraries are not **Load Modules**. designating: Designating something for a particular **Use** means specifying it for and restricting it to that **Use**. use: see item #64 above device class: The generic name for a group of device types. For example, all display stations belong to the same device class. A device class is different from a device type. A device type is composed of all devices that share a common model number or family (e.g. IBM 4331 printers).
73.	***digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security***	digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class: **Digitally Signing** a different ("second") **Load Module** by using a different ("second") **Digital Signature** as the signature **Key**, which signing indicates to any and all devices in the second device class that the signor authorized and restricted this **Load Module** for **Use** by that device. No **VDE** device can perform any execution of any **Load Module** without such authorization. The method ensures that the **Load Module** cannot execute in a particular device class and ensures that no device in that device class has the **Key(s)** necessary to verify the **Digital Signature**. All devices in the first device class have the same persistent (not just occasional) and identified level of **Tamper Resistance** and the same persistent and identified **Level of Security**. All devices in the second device class have the same persistent and

	'721 Claim 1	MS Construction
	level of the first device class;	identified level of **Tamper Resistance** and same persistent and identified **Level of Security**. The identified level of **Tamper Resistance** or identified **Level of Security** (or both) for the first device class, is greater than or less than the identified Level Of **Tamper Resistance** or identified **Level of Security** for the second device class. digital signature: see item #72 above designating: see item #72 above device class: see item #72 above load module: see item #72 above use: see item #64. level of security: An ordered measure of the degree of trustworthiness. The "security level" is persistent unless expressly noted to exist only some of the time. Also, the combination of a hierarchical classification and a set of nonhierarchical categories that represents the sensitivity of an object or the clearance of a subject. For example, Unclassified, Confidential, Secret, and Top Secret are hierarchical classifications, whereas NATO and NOFORN are non-hierarchical categories defined by the Department of Defense Trusted Computing guidelines. tamper resistance: The ability of a **Tamper Resistant Barrier** to prevent **Access**, observation, and interference with information or processing encapsulated by the barrier.
74.	*distributing the first load module for use by at least one device in the first device class*; and	distributing the first load module for use by at least one device in the first device class: The first **Load Module, Digitally Signed** as indicated above, is transmitted to at least one device in the first device class. load module: see item #72 above device class: see item #72 above
75.	*distributing the second load module for use by at least one device in the second device class.*	distributing the second load module for use by at least one device in the second device class: The second **Load Module, Digitally Signed** as indicated above, is transmitted to at least one device in the second device class. load module: see item #72 above device class: see item #72 above

	'721 Claim 34	MS Construction
76.	34. A protected processing environment comprising:	Claim as a Whole: The "**Protected Processing Environment**" is part of and within **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
77.	a first **tamper resistant barrier** having a first **security level**,	tamper resistant barrier: An active device that encapsulates and separates a **Protected Processing Environment** from the rest of the world. It prevents information and processes within the **Protected Processing Environment** from being observed, interfered with, and leaving except under appropriate conditions ensuring **Security**. It also **Controls** external access to the encapsulated **Secure** resources, processes and information. A **Tamper Resistant Barrier** is capable of destroying protected information in response to **Tampering** attempts. security level: see item #73 above
78.	a first **secure execution space**, and	secure execution space: An allocated **Portion** of the **Secure Memory** within a special-purpose secure processing unit which is isolated from the rest of the world, and protected from observation by (and encapsulated within) a **Tamper Resistant Barrier** and protected from alteration by the processor. The processor cryptographically verifies the integrity of all code loaded from **Secure Memory** prior to execution, executes only the code that the processor has authenticated for its use, and is otherwise **Secure**.
79.	at least one ***arrangement within the first tamper resistant barrier*** that	arrangement within the first tamper resistant barrier: An organization of hardware and software which arrangement is located and executed wholly within the first **Tamper Resistant Barrier**. arrangement: A collection of things that have been arranged. In context, the term requires an organization of hardware and software and data, or hardware and software, or hardware and data. tamper resistant barrier: see item #72 above
80.	***prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level.***	prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level: "A second secure execution space having a second tamper resistant barrier with a second security level different from the first security level": a second **Secure Execution Space** (different from the first **Secure Execution Space**) is part of the **Protected Processing Environment**, and has a **Tamper Resistant Barrier** (different from the first **Tamper Resistant Barrier**) that has a persistent (not just occasional) **Security Level** greater than or less than the first persistent **Security Level**. "The same executable accessed by": the same **Executable** (as opposed to, e.g., two copies of the same **Executable**) is simultaneously accessed by both the first **Secure Execution Space** and the second **Secure Execution Space**. "Prevents the first secure execution space from executing": the arrangement **Prevents** the first **Secure Execution Space**, otherwise capable of executing the **Executable**, from executing any part of the **Executable** (e.g., on behalf of any user, process, or device). prevents: Imposes an active restraint on an action such that it cannot occur by any means or under any circumstances. access (accessed): see item #29 above security level: see item #73 above

		secure execution space: see item #78 above tamper resistant barrier: see item #72 above

	'861 Claim 58	MS Construction
81.	58. A method of	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
82.	*creating a first secure container*, said method including the following steps;	creating a first secure container: This preamble language is a claim limitation. Completely forming (as opposed to defining) the **Secure Container**, within a **VDE Secure Processing Environment(s)**. secure container: see item #59 above
83.	**accessing a descriptive data structure**, said descriptive data structure ***including or addressing organization information at least in part describing a required or desired organization of a content section of said first secure container, and metadata information at least in part specifying at least one step required or desired in creation of said first secure container;***	including or addressing organization information at least in part describing a required or desired organization of a content section of said first secure container , and metadata information at least in part specifying at least one step required or desired in creation of said first secure container: The same single **Descriptive Data Structure** must either **Contain** within its confines or **Address** both **Organization Information** and **Metadata Information**. Both the "desired" organization of the content section and also the "desired" step, occur after the **Descriptive Data Structure** is accessed, not before. The Metadata Information explicitly dentifies a procedure ("step") that must be executed in creation of the first **Secure Container**, as opposed to **Iidentifying** a procedure to be run if later required or desired, as opposed to **Identifying** a result or a **Data Item** to be included in the first **Secure Container**, and as opposed to identifying information which operates as a parameter for a procedure. required: A condition without which an action cannot occur. A required condition acts prospectively – it does not apply to a description created at or after the creation of the object to which it applies. access (accessing): see item #29 above descriptive data structure: A machine-readable data structure (e.g., text file, template, etc.) **Containing** or **Addressing** descriptive information (e.g., **Metadata**, shorthand abstract representation, integrity constraints, Rules, instructions, etc.) about (1) the layout, generic format, attributes, or hierarchical structure of the contents section of one or a family of other data structure(s) (e.g., a rights management data structure), (2) the operations or processes used to create or **Use** such other data structure(s), and/or (3) the consequences of such operations. The **Descriptive Data Structure** is capable of being used to create or handle (e.g., read, locate information within, request information from, and/or manipulate) the other data structure(s). The **Descriptive Data Structure** is not **Associated With** the other data structure(s) and does not **Contain** or specify its particular contents (e.g., "Yankees Win the Pennant!"). addressing: Referring to something by the specific location where it is stored, without directly storing it. The location is explicitly identified by its name or number. Organization (organization, organization information): The manner in which data is represented and laid out in physical storage. For example, for data organized as records: the field hierarchy, order, type and size. organize: Representing and laying out data in a particular manner in physical storage. metadata information: Information that describes one or more attributes of other data, and/or the processes used to create and/or **Use** that data. For example, **Metadata Information** may describe the following attributes of other data: its meaning, representation in storage, what it is used for and by whom, context, quality and condition, location, ownership, or its data elements or their attributes (name, size, data

	'861 Claim 58	MS Construction
		type, etc.)
84.	**using said descriptive data structure to organize said first secure container contents;**	descriptive data structure: see item #83 above including: see item #2 above organize: see item #83 above
85.	using said metadata information to ***at least in part determine specific information required to be included in said first secure container contents***; and	at least in part determine specific information required to be included in said first secure container contents: The Metadata Information is used to determine the specific value, not merely the kind, of at least some of the information that must be placed inside the **Secure Container**. The use of the **Metadata Information** actively requires the **Secure Container** creation steps to add this specific information to the first **Secure Container**, as opposed to the specific information being within the **Secure Container** for some other reason. required: see item #83 above including (included): see item #2 above
86.	***generating or identifying*** *at least one* ***rule*** *designed to control at least one aspect of access to or use of at least a* ***portion*** *of said first secure container contents.*	generating or identifying at least one rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents: **Generating** or **Identifying Rule** designed for these particular **Secure Container** contents, which is used (by **VDE Control(s)** executing in **VDE Secure Processing Environment(s)**) to limit **Access** to or **Use** of at least a **Portion** of the contents of the first **Secure Container** (by all users, processes, and devices). Without compliance with this **Rule**, no process, user, or device is able to take the **Controlled** aspect of the **Controlled Access** or **Use** action. The **Rule** is generated or **Identified** based at least in part on the **Descriptive Data Structure**. generating: Producing. identifying: see item #19 above rule: see item #63 above control: see item #17 above aspect: see item #64 above access: see item #29 above use: see item #43 above portion: see item #9 above secure container: see item #59 above

	'891 Claim 1	MS Construction
87.	1. A method for using at least one	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
88.	***resource processed in a secure operating environment at a first appliance***, said method comprising:	resource processed in a secure operating environment at a first appliance: This preamble language is a claim limitation. A shared facility, required by a job or task, of a first appliance's **Secure Operating Environment** which is processed within that **Secure Operating Environment**'s special-purpose Secure Processing Unit. A Secure Processing Unit is a special-purpose unit isolated from the rest of the world in which a hardware **Tamper Resistant Barrier** encapsulates a processor and internal **Secure Memory**. The **Tamper Resistant Barrier** prevents all unauthorized interference, removal, observation, and **Use** of the information and processes within it. The processor cryptographically verifies the integrity of all code loaded from the **Secure Memory** prior to execution, executes only the code that the processor has authenticated for its **Use**, and is otherwise **Secure**. resource processed: A record containing control information, which record is stored and acted upon within a processing environment. secure operating environment: Same as **Secure Processing Environment**.
89.	***securely receiving a first entity's control at said first appliance***, said first **entity** being located remotely from said **operating environment** and said first appliance;	securely receiving a first entity's control at said first appliance: This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("**Securely** receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b). The recited function requires: A first appliance obtaining a **VDE Secure Container** encapsulating a **Control** created, selected, or modified by a first entity, as part of a communication encrypted on the communications level, authenticating the first appliance in accordance with **VDE Controls Associated With** the **Secure Container**, and accepting the **Secure Container**. entity: Any person or organization. entity's control: **Control** created, modified, or selected by any person or organization to **Control** a particular **Use** of or **Access** to particular **Protected** information by a particular user(s). control: see item #17 above operating environment: see item #88 above
90.	***securely receiving a second entity's control at said first appliance***, said second entity being located remotely from said operating environment and said first appliance, said second entity being different from said first entity; and	securely receiving a second entity's control at said first appliance: This claim language falls within 35 U.S.C. § 112, 6. It recites a step or result ("securely receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b). The recited function requires: A first appliance obtaining a **VDE Secure Container** encapsulating a Control created, selected, or modified by a second entity, as part of a communication encrypted on the communications level, **Authenticating** the first appliance in accordance with **VDE Controls Associated With** the **Secure Container**, and accepting the **Secure Container**.

	'891 Claim 1	MS Construction
		entity's control: see item #89 above control: see item #17 above
91.	***securely processing a data item at said first appliance, using at least one resource,*** including	securely processing a data item at said first appliance, using at least one resource, including : Performing an operation, inside the special-purpose Secure Processing Unit of the first appliance, on a **Data Item** inside the Secure Processing Unit. The operation cannot be observed from outside the Secure Processing Unit and is performed only after the integrity of the program code for performing such operation is cryptographically verified. A Secure Processing Unit is a special-purpose unit isolated from the rest of the world in which a hardware **Tamper Resistant Barrier** encapsulates a processor and internal **Secure Memory**. The **Tamper Resistant Barrier** prevents all unauthorized interference, removal, observation, and Use of the information and processes within it. The processor cryptographically verifies the integrity of all code loaded from the **Secure Memory** prior to execution, executes only the code that the processor has authenticated for its Use, and is otherwise **Secure**. control: see item #17 above data item: An individual unit of digital information representing a single value, such as that stored in a field of a larger **Record** in a database. It is the smallest useful unit of named information in the system. resource: A shared facility of a computing system or operating system, which is required by a job or task, and is processed by a processing unit.
92.	***securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item.***	securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item: Processing the resource (component part of a first appliance's **Secure** Operating Environment) within the **Secure** Operating Environment's special-purpose Secure Processing Unit to execute the first Control and second Control in combination within the Secure Processing Unit. This execution of these **Controls Governs** all Use of the **Data Item** by all users, processes, and devices. The processing of the **Resource** and execution of the **Controls** cannot be observed from outside the Secure Processing Unit and is performed only after the integrity of the **Resource** and **Controls** is cryptographically verified. A Secure Processing Unit is a special-purpose unit isolated from the rest of the world in which a hardware **Tamper Resistant Barrier** encapsulates a processor and internal **Secure Memory**. The **Tamper Resistant Barrier** prevents all unauthorized interference, removal, observation, and Use of the information and processes within it. The processor cryptographically verifies the integrity of all code loaded from the **Secure Memory** prior to execution, executes only the code that the processor has authenticated for its Use, and is otherwise **Secure**. control: see item #17 above data item: see item #91 above resource: see item #91 above use: see item #43 above govern: see Control (v.) item #7 above

	'900 Claim 155	MS Construction
93.	155. A **virtual distribution environment** comprising	Claim as a Whole: The "virtual distribution environment" is **VDE**. VDE/Virtual Distribution Environment: Data Security and Commerce World: InterTrust's February 13, 1995, patent application described as its "invention" a **Virtual Distribution Environment** ("**VDE** invention") for **Securing**, administering, and auditing all **Security** and commerce digital information within its multi-node world (community). **VDE** guarantees to all **VDE** "participants" identified in the patent application that it will limit all **Access** to and **Use** (i.e., interaction) of such information to **Authorized** activities and amounts, will ensure any requested reporting of and payment for such **Use**, and will maintain the availability, secrecy, integrity, non-repudiation and authenticity of all such information present at any of its nodes (including **Protected** content, information about content usage, and content **Controls**.). **VDE** is **Secure** against at least the threats identified in the Feburary 1995, patent application to this availability (no user may delete the information without **Authorization**), secrecy (neither available nor disclosed to unauthorized persons or processes), integrity (neither intentional nor accidental alteration), non-repudiation (neither the receiver can disavow the receipt of a message nor can the sender disavow the origination of that message) and authenticity (asserted characteristics are genuine). **VDE** further provides and requires the components and capabilities described below. Anything less than or different than this is not **VDE** or the described "invention." Secure Processing Environment: At each node where **VDE-Protected** information is **Accessed**, **Used**, or assigned control information, **VDE** requires a **Secure Processing Environment**. A **Secure Processing Environment** is uniquely identifiable, self-contained, non-circumventable, and trusted by all other **VDE** nodes to protect the availability, secrecy, integrity and authenticity of all information identified in the patent application as being **Protected**, and to guarantee that such information will be **Accessed** and **Used** only as expressly **Authorized** by the associated **VDE Controls**, and to guarantee that all requested reporting of and payments for protected information use will be made. A **Secure Processing Environment** is formed by, and requires, a Secure Processing Unit having a hardware **Tamper Resistant Barrier** encapsulating a processor and internal **Secure Memory**. The **Tamper Resistant Barrier** prevents all unauthorized interference, removal, observation, and other **Use** of the information and processes within it. VDE Controls: **VDE Allows Access** to or **Use** of **Protected** information and processes only through execution of (and satisfaction of the requirements imposed by) independent, special-purpose, **Executable VDE Control(s)**. Each **VDE Control** is a **Component Assembly** dedicated to a particular activity (e.g., editing, modifying another **Control**, a user-defined action, etc.), particular user(s), and particular protected information. Each separate information **Access** or **Use** is independently **Controlled** by independent **VDE Control(s)**. A **VDE Control** can execute only within a **Secure Processing Environment**. Each **VDE Control** is assembled, within a **Secure Processing Environment**, from independently deliverable modular components (e.g., **Load Modules** or other **Controls**), dynamically in response to an information **Access** or **Use** request. The dynamic assembly of a **Control** is directed by a "blueprint" **Record** (put in place by one or more **VDE** users) **Containing** control information identifying the exact modular code components to be assembled and executed to **Govern** this particular activity on this particular information by this particular user(s). Each **Control** is independently assembled, loaded and delivered vis-à-vis other **Controls**. Control information and **Controls** are extensible and can be configured and modified by all users, and combined by all users with any other **VDE** control information or **Controls** (including that provided by other users), subject only to "senior" user **Controls**. Users can assign control information and **Controls** to all of or an arbitrarily fine, user-defined **Portion** of the **Protected** information, such as a single paragraph of a document, as opposed to being limited to file-based controls.

	'900 Claim 155	MS Construction
		VDE Controls reliably limit **Access** and **Use** of the protected information to **Authorized** activities and amounts. VDE Secure Container: A **VDE Secure Container** is a self-contained, self-protecting data structure which (a) encapsulates information of arbitrary size, type, format, and organization, including other, nested, containers, (b) cryptographically protects that information from all unauthorized **Access** and **Use**, (c) provides encrypted storage-management functions for that information, such as hiding the physical storage location(s) of its **Protected** contents, (d) permits the **Association** of itself and/or all of or arbitrary **Portions** of its contents with **Controls** and control information **Governing Access** to and **Use** thereof, and (e) **Prevents** such **Use** or **Access** (as opposed to merely **Preventing** decryption) until it is opened. A **Secure Container Can Be** opened only as expressly **Allowed** by the **associated VDE Control(s)**, only within a **Secure Processing Environment**, and only through decryption of its encrypted header. A **Secure Container** is not directly accessible to any non-**VDE** calling process. All such calls are intercepted by **VDE**. The creator of a **Secure Container** can assign (or allow others to assign) control information to all of or any arbitrary **Portion** of a **Secure Container's** contents, or to an empty **Secure Container** (to **Govern** the addition of contents to the **Secure Container**, and **Access** to or **Use** of those contents). A container is not a **Secure Container** merely because its contents are encrypted and signed. All **VDE-Protected** information (including protected content, information about content usage, and **Controls**) is encapsulated within a **Secure Container** whenever stored outside a **Secure Processing Environment** or **Secure Database**. Non-Circumventable: **VDE** is non-circumventable (sequestered). It intercepts all attempts by any and all users, processes, and devices, to **Access** or **Use**, such as observing, interfering with, or removing) **Protected** information, and **Prevents** all such attempts other than as **Allowed** by execution of (and satisfaction of all requirements imposed by) **Associated VDE Controls** within **Secure Processing Environment(s)**. Peer to Peer: **VDE** is peer-to-peer. Each **VDE** node has the innate ability to perform any role identified in the patent application (e.g., end user, content packager, distributor, **Clearinghouse**, etc.), and can protect information flowing in any direction between any nodes. **VDE** is not client-server. It does not pre-designate and restrict one or more nodes to act solely as a "server" (a provider of information (e.g., authored content, control information, etc.) to other nodes) or "client" (a requestor of such information). All types of protected-content transactions can proceed without requiring interaction with any server. Comprehensive Range of Functions: **VDE** comprehensively **Governs** all **Security** and commerce activities identified in the patent application, including (a) metering, budgeting, monitoring, reporting, and auditing information usage, (b) billing and paying for information usage, and (c) negotiating, signing and enforcing contracts that establish users' rights to **Access** or **Use** information. User-Configurable: The specific protections **Governing** specific **VDE-Protected** information are specified, modified, and negotiated by **VDE's** users. For example, **VDE** enables a consumer to place limits on the nature of content that may be accessed at her node (e.g., no R-rated material) or the amount of money she can spend on viewing certain content, both subject only to other users' senior **Controls**. General Purpose; Universal: **VDE** is universal as opposed to being limited to or requiring any particular type of appliance, information, or commerce model. It is a single, unified standard and environment within which an unlimited range of electronic rights protection, data **Security**, electronic currency, and banking applications can run. Flexible: **VDE** is more flexible than traditional information **Security** and commerce

	'900 Claim 155	MS Construction
		systems. For example, **VDE** allows consumers to pay for only the user-defined **Portion** of information that the user actually uses, and to pay only in proportion to any quantifiable **VDE** event (e.g., for only the number of paragraphs displayed from a book), and allows editing the content in **VDE** containers while maintaining its **Security**.
94.	*a first host processing environment comprising*	a first host processing environment comprising: A **Host Processing Environment** that encompasses the recited computer hardware (central processing unit, main **Memory**, and mass storage) and certain **VDE Protected Processing Environment** software loaded in that main **Memory** and executing in that central processing unit, but does not encompass software, such as the recited **Tamper Resistant Software**, which is stored in mass storage and not executing. host processing environment: A processing environment within a **VDE** node which is not a **Secure Processing Environment**. A "host processing environment" may either be "secure" or "not secure." A "secure host processing environment" is a self-contained **Protected Processing Environment**, formed by loaded, **Executable** programming executing on a general purpose CPU (not a Secure Processing Unit) running in protected (privileged) mode. A "non-secure host processing environment" is formed by loaded, **Executable** programming executing on a general purpose CPU (not a Secure Processing Unit) running in user mode.
95.	a central processing unit;	
96.	main **memory** operatively connected to said central processing unit;	memory: see item #3 above
97.	mass storage operatively connected to said central processing unit and said main **memory**;	memory: see item #3 above
98.	*said mass storage storing **tamper resistant software***	said mass storage storing tamper resistant software: The **Tamper Resistant Software** is physically stored within, as opposed to being merely **Addressed** by, the mass storage. tamper resistant software: Software that is encapsulated and executed wholly within a **Tamper Resistant Barrier**.
99.	*designed to be loaded into said main memory and executed by said central processing unit,*	designed to be loaded into said main memory and executed by said central processing unit: The **Tamper Resistant Software** is capable of being loaded into only said main **Memory** and is capable of being executed only by said central processing unit.
100.	*said tamper resistant software comprising: **machine check programming** which **derives** information from one or more **aspects** of said **host processing environment**, one or more **storage locations** storing said information;*	said tamper resistant software comprising: machine check programming which derives information from one or more aspects of said host processing environment, one or more storage locations storing said information: The **Tamper Resistant Software** within said mass storage includes one or more storage locations within it. These storage locations are designated to store, and must store, information **Derived** by the **Machine Check Programming**, and must not store any other information. machine check programming: **Executable** programming that when executed checks a machine and generates a unique "machine signature" which distinguishes the physical machine from all other machines. This machine check programming code sometimes is invoked by integrity programming. host processing environment: see item #94 above

	'900 Claim 155	MS Construction
		derives: To retrieve from a specified source. aspect: see item #64 above
101.	*derives information from one or more aspects of said host processing environment*	derives information from one or more aspects of said host processing environment: **Deriving** from the **Host Processing Environment** hardware one or more values that uniquely and persistently **Identify** the **Host Processing Environment** and distinguish it from other **Host Processing Environments.** The "one or more aspects of said host processing environment" are persistent elements or properties of the **Host Processing Environment** itself that are capable of being used to distinguish it from other environments, as opposed to, e.g., data or programs stored within the mass storage or main **Memory**, or processes executing within the **Host Processing Environment.** host: see item #94 above derives: see item #100 above aspect: see item #64 above
102.	*one or more storage locations storing said information;*	One or more storage locations: One or more logical storage locations within the Tamper Resistant Software storing only information Derived by the Machine Check Programming.
103.	**integrity programming** which causes said machine check programming to **derive** said information, **compares** said information to *information previously stored in said one or more storage locations*, and	integrity programming: Executable programming that when executed checks and reports on the integrity of a device or process. "Integrity" means the property that information has not been altered either intentionally or accidentally. information previously stored in said one or more storage locations: Any information once stored in said "one or more storage locations storing said information," but not stored therein when the recited comparison occurs. information previously stored: Information that once was stored but is no longer stored. derive: see item #100 above compares: A processor operation that evaluates two quantities and sets one of three flag conditions as a result of the comparison – greater than, less than, or equal to.
104.	*generates an indication based on the result of said comparison*; and	generates an indication based on the result of said comparison: Producing an indication based on the result of the "compares" step. The "indication" need not be displayed to a user. The indication is based solely on that result. There are only two possible indications: exact match found or exact match not found. comparison: see item #103 above
105.	*programming which takes one or more actions based on the state of said indication;*	programming which takes one or more actions based on the state of said indication: **Executable** programming code that is a part of the Tamper Resistant Software, when executed, and not a part of the **Host Processing Environment**. Whenever the recited indication is generated, no matter what it indicates, this code (executing on the CPU for which it was designed and loaded in the **Memory** for which it was designed) must take an action, or more than one action. The particular action(s) taken must be based

	'900 Claim 155	MS Construction
		solely on the state of that indication.
106.	said one or more actions including *at least temporarily halting further processing.*	at least temporarily halting further processing: The action(s) taken by this programming must encompass **Halting** or temporarily **Halting** all further processing of the **Host Processing Environment** and any processes running within it. halting: Stopping execution of a running (executing) process unconditionally (i.e., without providing any specific condition for resumption). For example, executing an instruction known as a "breakpoint halt instruction."

	'912 Claim 8	MS Construction
107.	8. A process comprising the following steps:	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
108.	accessing a first **record** containing information directly or indirectly **identifying** one or more elements of a first **component assembly**,	record: A data structure that is a collection of fields (elements), each with its own name and type. Unlike an array, whose elements are accessed using an index, the elements of a record are accessed by name. A record can be accessed as a collective unit of elements, or the elements can be accessed individually. identifying: see item #19 above access: see item #29 above comparison: see item #103 above component assembly: A cohesive **Executable** component created by a channel which binds or links together two or more independently deliverable **Load Modules**, and **Associated** data. A **Component Assembly** is assembled, and executes, only within a **VDE Secure Processing Environment**. A **Component Assembly** is assembled dynamically in response to, and to service, a particular content-related activity (e.g., a particular **Use** request). Each **VDE Component Assembly** is assigned and dedicated to a particular activity, particular user(s), and particular **Protected** information. Each **Component Assembly** is independently assembled, loadable and deliverable vis-à-vis other **Component Assemblies**. The dynamic assembly of a **Component Assembly** is directed by a "blueprint" **Record Containing Control** information for this particular activity on this particular information by this particular user(s). **Component Assemblies** are extensible and can be configured and reconfigured (modified) by all users, and combined by all users with other **Component Assemblies**, subject only to other users' "senior" **Controls**.
109.	at least one of said elements **including** at least some **executable programming**,	executable programming: Executable: A cohesive series of machine code instructions in a format that can be loaded into **Memory** and run (executed) by a connected processor. executable programming: A cohesive series of machine code instructions, comprising a computer program, in a format that can be loaded into **Memory** and run (executed) by a connected processor. (A "computer program" is a complete series of definitions and instructions that when executed on a computer will perform a required or requested task.) including: see item #2 above
110.	at least one of said elements constituting a **load module**,	load module: see item #72 above
111.	said **load module** including **executable programming** and a header;	load module: see item #72 above including: see item #2 above executable programming: see item #109 above
112.	said header including an **execution space identifier** ***identifying at least one aspect of an execution space required for use***	identifying at least one aspect of an execution space required for use and/or execution of the load module: Defining fully, without reference to any other information, at least one of the persistent elements or properties (aspects) (that are capable of being used to distinguish it from other environments of an **Execution Space**) that are **Required** for any **Use**, and/or for any execution, of the **Load Module**. An **Execution Space** without

	'912 Claim 8	MS Construction
	and/or execution of the load module **associated with** said header;	all of those **Required** aspects is incapable of making any such execution and/or other **Use** (e.g., **Copying**, displaying, printing) of the **Load Module**. including: see item #2 above execution space identifier: A value that uniquely identifies a particular execution space. execution space: A processor-addressable physical **Memory** into which data and **Executable** code can be loaded, which is assigned to a single executing process while that process is actively executing. **Memory** holding "swapped out" processes or **Executables** is not part of an "execution space." load module: see item 110 above required: see item #83 above aspect: see item #64 above associated with: see item #4 above identifying: see item #19 above
113.	***said execution space identifier provides the capability for distinguishing between execution spaces providing a higher level of security and execution spaces providing a lower level of security;***	said execution space identifier provides the capability for distinguishing between execution spaces providing a higher level of security and execution spaces providing a lower level of security: The **Execution Space Identifier**, by itself, provides the **Load Module** with the capability of determining the persistent **Level of Security** of any Execution Space in which it is loaded, and of distinguishing between any two Execution Spaces based on their respective, determined persistent (not just occasional) "Levels Of Security." This capability extends to at least two Execution Spaces providing a higher **Level of Security** and at least two Execution Spaces providing a lower **Level of Security.** execution space identifier: see item #112 above execution space: see item #112 above level of security: see **Security Level**, item #73 above
114.	using said information to **identify** and locate said one or more elements;	identify: see item #19 above
115.	accessing said located one or more elements;	access: see item #29 above
116.	**securely assembling** said one or more elements to form at least a portion of said first **component assembly**;	securely assembling: **Securely** (1) linking or binding plural distinct elements together in a particular manner (specified by authenticated assembly instructions) into a single cohesive **Executable** unit so the elements can directly reference each other element within the resulting assembly, within a **VDE Secure Processing Environment**, (2) validating and verifying the authenticity and integrity of each element (e.g., that it has not been modified from or substituted for the correct element) immediately prior to binding it into the assembly, and (3) ensuring that the elements are linked together only in ways that are intended by the **VDE** participants who created the elements and/or specified the assembly thereof. component assembly: see item #108 above
117.	executing at least some of said **executable programming**; and	executable programming: see item #109 above

	'912 Claim 8	MS Construction
118.	***checking said record for validity prior to performing said executing step.***	checking said record for validity prior to performing said executing step: Before executing any **Executable Programming** encompassed within any element which is directly or **indirectly identified by any information Contained** within the first **Record**, evaluating, within a **VDE Secure Processing Environment**, the values and formats of all data fields within the first **Record** and confirming that they have legitimate values and formats. record: see item #108 above validity: The state in which authenticated data conforms to predetermined completeness and consistency parameters.

'912 Asserted Claim 35

	'912 Claim 35	MS Construction
119.	35. A process comprising the following steps:	Claim as a whole: The recited method is performed within a **VDE**. (See item #93 for Microsoft's construction of **VDE**.)
120.	at a first **processing environment** receiving a first **record** from a second processing environment remote from said first processing environment;	processing environment: A standardized, well-defined, self-contained, computing base, formed by hardware and executing code, that provides an "interface" and set of resources which can support different applications, on different types of hardware platforms. In the context of claim 35 of the '912 patent: a **Secure Processing Environment**. record: see item #108 above
121.	said first record being *received in a secure container*;	received in a secure container: The first **Processing Environment** obtained a **VDE Secure Container** encapsulating the **Record** inside, and authenticated the intended recipient in accordance with **VDE Controls Associated With** the **Secure Container**, and accepted the **Secure Container**. secure container: see item #59 above
122.	said first record **containing** identification information directly or indirectly **identifying** one or more elements of a first **component assembly**;	containing: see item #60 above identifying: see item #19 above component assembly: see item #108 above
123.	at least one of said elements **including** at least some executable programming;	including: see item #2 above
124.	*said component assembly allowing access to or use of specified information;*	said component assembly allowing access to or use of specified information: The **Component Assembly** identifies specific information (the specific value, not merely the kind of information) over which it (by itself and with no other information), executing in a **VDE Secure Processing Environment**, **Allows Access** or **Use** (as opposed to **Access** "and" **Use**). Unless **Allowed** by the **Component Assembly**, no user, process, or device is able to **Access** or **Use** the specified information. The **Component Assembly** is **Associated With** and dedicated to this particular specified information. component assembly: see item #108 above allow (allowing): see item #10 above access: see item #29 above
125.	said **secure container** also **including** a first of said elements;	secure container: see item #59 above including: see item #2 above
126.	**accessing** said first **record**;	access: see item #29 above record: see item #108 above
127.	using said identification information to **identify** and locate said one or more elements;	identify: see item #19 above

128.	said locating step including locating a second of said elements at a third processing environment located remotely from said first **processing environment** and said second **processing environment**;	processing environment: see item #120 above
129.	accessing said located one or more elements;	access (accessing): see item #29 above
130.	said element accessing step including retrieving said second element from said third processing environment;	
131.	**securely assembling** said one or more elements to form at least a portion of ***said first component assembly specified by said first record***; and	said first component assembly specified by said first record: The first **Record** by itself **Contains** sufficient information to unambiguously **Identify** the assembled **Component Assembly**, including all of its elements. This limitation is inconsistent with the recitation "first record containing identification information directly or indirectly identifying one or more elements of first component assembly." securely assembling: see item #116 above component assembly: see item #108 above record: see item #108 above
132.	executing at least some of said **executable programming,**	executable programming: see item #109 above
133.	said executing step taking place at said first **processing environment.**	processing environment: see item #120 above